KEYBOARD-ONLY WALKTHROUGH NOTES

1. Tab from page load → focus moves to “Skip to main content”

HTML that enables this

```
<a href="#maincontent">Skip to main content</a>
```

How it works

- When a web page loads, the browser sets the starting point of keyboard navigation at the very beginning of the document.
- The Tab key moves focus to the first *focusable* element in the DOM.
- An <a> element with an href is natively focusable because it is an interactive HTML element defined in browser standards.
- Therefore, if the skip-link is the first focusable element in the HTML source, it naturally becomes the first keyboard stop when pressing Tab.

Expected behavior

- User presses **Tab** → browser highlights/selects the skip link.
- No scripts or styles are necessary because this is built into HTML semantics.

2. Enter on skip link → focus + scroll moves to <main>

HTML that enables this

```
<main id="maincontent">
  <!-- main content here -->
</main>
```

How it works

- When a user activates a link with href="#some-id", the browser automatically performs two actions:
    1. Scrolls to the element with the matching ID.
    2. Moves focus to that element if it is a recognized sectioning element or can receive focus.
- The <main> element is an HTML5 landmark. Browsers treat it as a semantic section meant for primary content.
- Because of that, when href="#maincontent" is triggered, the browser:
    - Locates <main id="maincontent">.
    - Scrolls the viewport so that this element is in view.
    - Shifts keyboard focus to it as part of link-fragment behavior.

Expected behavior

- Pressing Enter on the skip-link should immediately jump the page visually and move focus to the main content area.

3. **Tabbing moves through navigation links in logical order**

HTML that enables this

```
<nav>
  <a href="/about">About</a>
  <a href="/services">Services</a>
  <a href="/contact">Contact</a>
</nav>
```

How it works

- All anchors with an href attribute are placed into the tab order automatically by the browser.
- The order is determined by the physical order of elements in the HTML, known as DOM source order.
- This means the browser follows a top-to-bottom, left-to-right reading pattern, matching how content is structured in the markup.
- Keyboard users rely on this consistent sequence to understand page structure.

Expected behavior

- Pressing Tab cycles through:
    1. Home →About→Consult→Contact

4. **Reaching “Full name” via Tab, typing into it, pressing Tab to reach “Email”**

HTML that enables this

```
<form>
  <label for="fullname">Full name</label>
  <input id="fullname" name="fullname" type="text" required>

  <label for="email">Email</label>
  <input id="email" name="email" type="email" required>

  <button type="submit">Submit</button>
</form>
```

How it works

- Inputs (<input>) and buttons (<button>) have built-in keyboard focusability defined by the HTML specification.
- When the user presses the Tab key:
    - Focus moves to the next focusable element as determined by DOM order.
- Typing works because text-based input elements are designed to:
    - Accept text characters when they have focus.
    - Display a blinking caret automatically, without scripts.

Expected behavior

- Tab → Full name input
- Typing is possible because the input is now active
- Tab again → Email input

5. **Submitting the form with empty email → browser shows native validation message**

HTML that triggers validation

```
<input id="email" name="email" type="email" required>
```

How it works

- HTML5 introduced built-in form validation for browsers.
- Two attributes control the behavior:
    1. type="email"
        - Tells the browser that the input's value must contain a valid email format (must include @, no spaces, etc.)
    2. required
        - Tells the browser the field cannot be empty.
- When the user activates the Submit button (either by clicking or pressing Enter):
    - The browser checks each field against its HTML-defined constraints.
    - If a field fails a rule (empty + required, bad email format), the browser:
        - Stops form submission.
        - Shows a native error message popup.
        - Highlights the invalid field.

Expected behavior

- Pressing Enter on Submit with empty email should show messages like:
    - "Please fill out this field."
    - "Please enter an email address."

6. **Shift+Tab moves focus backward in reverse order**

HTML that enables this

(No special code required; any meaningful HTML structure works.)

```
<a href="#maincontent">Skip to main content</a>

<nav>
  <a href="/about">About</a>
  <a href="/services">Services</a>
  <a href="/contact">Contact</a>
</nav>

<form>
  <label for="fullname">Full name</label>
  <input id="fullname" type="text">
  <label for="email">Email</label>
  <input id="email" type="email">

  <button type="submit">Submit</button>
</form>
```

How it works

- Browsers maintain not only a forward tab order but a reverse order as well.
- When the user presses:
  - Tab → moves forward in the order elements appear in the HTML.
  - Shift + Tab → moves backward through the same sequence.
- This behavior is embedded into the browser’s keyboard navigation algorithm and applies automatically to all focusable HTML elements.

Expected behavior

- The user can move backward through:
  - Submit → Email → Full name → Contact → Services → About → Skip link
- Without needing CSS, JS, or ARIA.